Notebook : Préprocessing → Modélisation → Métriques

Ce notebook applique :

- Encodage auto (One-Hot + Label Encoding si nécessaire)
- Pipeline scikit-learn
- Test de plusieurs modèles

In [ ]:
```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.ensemble import RandomForestRegressor
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor

df = pd.read_csv('/mnt/data/dataset_building_cleanful.csv')

drop_cols = [
    "OSEBuildingID",
    "TaxParcelIdentificationNumber",
    "PropertyName",
    "Address",
    "City",
    "State",
    "ZipCode",
    "ListOfAllPropertyUseTypes"
]

df = df.drop(columns=drop_cols)

target = "SiteEUIWN(kBtu/sf)"

y = df[target]
X = df.drop(columns=[target])

cat_cols = [
    "BuildingType",
    "PrimaryPropertyType",
    "LargestPropertyUseType",
    "ComplianceStatus"
]

num_cols = [c for c in X.columns if c not in cat_cols]

preprocessor = ColumnTransformer(
    transformers=[
        ("num", StandardScaler(), num_cols),
        ("cat", OneHotEncoder(handle_unknown="ignore"), cat_cols)
    ]
)

models = {
    "LinearRegression": LinearRegression(),
    "DecisionTree": DecisionTreeRegressor(),
    "RandomForest": RandomForestRegressor()
}

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta

results = {}

for name, model in models.items():
    pipe = Pipeline([
        ("preproc", preprocessor),
        ("model", model)
    ])
    pipe.fit(X_train, y_train)
    pred = pipe.predict(X_test)
    results[name] = {
        "RMSE": mean_squared_error(y_test, pred, squared=False),
        "R2": r2_score(y_test, pred)
    }

results
```